附录 WorkBuddy 100 问

A.1 基础认识与上手（01—10）

01. WorkBuddy 和普通聊天 AI 有什么不同？

答：WorkBuddy 可以理解为面向完整工作任务的智能体工作台。普通聊天 AI 通常停留在解释、建议或生成一段文字；WorkBuddy 在获得授权后，可以围绕目标读取工作目录中的资料，规划步骤，调用 Skill、连接器或其他工具，并生成文档、表格、演示稿等产物。因此，评价一次任务不能只看回复是否流畅，而要看它读了哪些资料、执行了哪些动作、产物保存在哪里、能否打开修改，以及关键事实能否回到原始材料核查。最后仍由人验收，未通过就明确指出问题再返工。

02. 新手第一次应该让 WorkBuddy 做什么？

答：第一次使用时，不要搭复杂智能体，而是创建独立工作目录，完成一个输入和输出都清楚的小任务。可以把一份练习会议记录放进 input 目录，要求生成结构化纪要和待办清单；也可以先让它为文件列清单和重命名建议，但在确认前不移动文件。执行时观察计划、工具调用和文件变化，完成后在结果区预览并打开产物。这样一轮就能同时验证目录授权、文件读取、任务表达、结果保存和人工验收，比一上来尝试自动化更容易定位问题。完成后把任务文本和结果一起保存，方便下次复用。

03. 为什么不建议第一次就处理敏感资料？

答：工作目录既是效率设置，也是安全边界。第一次使用时，你还不知道应用实际获得了哪些目录权限，生成文件会落在哪里，是否可能覆盖同名文件，也不熟悉分享或外部连接的范围。先用公开、虚构或脱敏材料建立演练目录，并把原件、过程区和输出区分开。确认它只读取授权范围、不会改动原件、产物可追溯后，再处理真实业务资料。客户信息、合同、财务、账号凭证和未公开内容，还应继续遵守单位的数据与权限制度。首次演练完成后，再逐级扩大资料和权限范围。

04. 什么叫“任务闭环”？

答：一个完整的任务闭环是：先说清目标，随后读取已授权资料，拆解任务并选择工具，执行后生成可检查的产物，再由人验收；不合格就带着具体意见返工，通过后才归档或发布。闭环的重点是每一步都有证据：输入文件明确、操作范围可控、结果文件有路径、事实能找到来源、缺失项被标出。只在对话框里得到一段看似完整的回答，还不能算完成；文件打不开、数字无法核对、责任人被凭空补全，也都说明闭环没有真正结束，还需回到对应步骤修正。

05. WorkBuddy 最适合解决哪类工作？

答：最适合 WorkBuddy 的工作通常具备三点：目标能说清、材料能提供、结果能检查。例如整理文件前先生成清单，依据会议转写制作纪要，把 Word 材料变成可编辑 PPT，对 Excel 做清洗和分析，或从多个信息源形成带来源的调研结果。它也能处理路径不完全确定的复杂任务，但需要计划和中间确认。相反，如果目标只是“帮我想想”、资料口径混乱、成功标准没人能判断，即使模型输出很多内容，也很难成为稳定交付。先选一项小任务验证，比罗列功能更有意义。

06. 哪些事情不能完全交给 WorkBuddy？

答：大模型擅长理解、归纳、改写和生成，但不天然保证事实正确，也不知道企业最新内部状态，更不承担业务与法律责任。法律、财务、医疗、人事、投资等高影响场景，可以让 WorkBuddy

收集材料、拆分问题、生成草稿或列出风险，却不能把它的结论直接当作最终决定。对外发布、付款、删除、数据库写入和制度变更等动作，也应设置人工确认。判断是否越界的简单方法是：一旦做错会造成不可逆损失，就必须由有权限的人复核并拍板。AI 可以辅助判断，但不能替你承担责任。

07. “一句话让 AI 干活”是不是越短越好？

答：可以把 Prompt 理解为“交给同事的任务单”，而不是越短越灵的咒语。简单查询确实可以一句话，但只要任务涉及文件、格式或真实操作，就应写清目标、输入、动作、约束、输出和验收六项。例如“分析销售表并做 PPT”还不够，还要说明哪份表是唯一口径、受众是谁、页数和结构、输出路径，以及数字必须与源文件一致。所谓“一句话干活”，真正有效的是一句话里信息完整，而不是把关键判断都丢给模型猜；执行前还要看计划是否理解正确。

08. WorkBuddy 能直接替代专业人员吗？

答：更准确的定位是把 WorkBuddy 当作能执行任务的协作者，而不是替代责任主体。一次 AI 工作可以分成模型、Agent、Skill、工具与人：模型负责理解和生成，Agent 组织行动，Skill 提供方法，工具触达文件和系统，而人负责边界与验收。它可以显著减少查找、搬运、整理、起草和格式处理，但专业口径、品牌方向、法律判断、财务结论仍需专业人员确认。名字里出现“专家”也不等于自动获得资质，关键结论必须查看来源、假设和不确定项。

09. 如何判断一个任务适不适合用 WorkBuddy？

答：可以先做一轮六问筛选：最终要解决什么问题，依据哪些文件或系统，需要分析、整理还是写入，有哪些不能碰的边界，交付什么文件，以及谁按什么标准验收。这也是任务卡和自动化设计的共同基础。若这六项大多能回答，任务通常适合执行；若目标清楚但路径不清，可以先用计划或 Agent 探索；若连资料口径、输出对象和成功标准都没有，则应先澄清需求。尤其在不可逆操作中，不能以“模型应该懂”为理由跳过确认，还要提前写明停止条件和责任人。

10. 新手学习 WorkBuddy 最有效的方法是什么？

答：更有效的学习路径是：先围绕真实小任务形成闭环，再逐步固化。第一阶段练文件读取、摘要、改写、表格提取和会议纪要；每次保存原始任务文本、输入样例、生成文件、验收问题和修改记录。第二阶段把反复成功的任务整理成任务卡，补上异常处理。只有同类任务稳定跑过三五次，才考虑写成 SOP、Skill 或自动化。这样学到的不是按钮位置，而是如何把工作拆成可执行、可验证、可复用的系统，并逐步形成自己的工作标准。每周复盘一次，比集中看大量教程更有效。

A.2 任务表达与拆解（11—20）

11. 一个好任务至少要写清楚什么？

答：一个完整任务包含六个要素：目标、输入、动作、约束、输出和验收。目标说明最终解决什么问题；输入指出使用哪些文件、目录、链接或系统；动作说明是分析、整理、转换还是写入；约束列出不能修改的内容和必须遵守的规则；输出写清文件类型、命名和保存位置；验收则规定怎样判断合格。正式办公任务还可补充受众、使用场合和截止时间。六项不一定都写成长段，但不能把关键决策留给模型自行猜测；任务越正式，字段越要明确且可核对。发送前可用这六项快速自查。

12. “帮我整理一下”为什么容易得到空泛结果？

答：因为“整理”不是一个可验收的结果。实际使用中，同一份资料可以被整理成摘要、目录、文件清单、会议纪要、待办表或演示稿，不同产物需要完全不同的步骤。更有效的写法是指定处理对象、动作、字段和停止条件，例如“从会议记录提取结论、待办、负责人、日期和待确认问题；不能确认的字段写待确认，不得补全”。如果涉及文件移动，还应先产出 inventory 和 proposed-actions，得到确认后再执行，避免把含糊语言变成真实误操作。

13. 怎样让输出更符合自己的使用场景？

答：先回答“这份材料要帮助谁做什么决定”。受众和场合会直接改变信息结构：管理层通常需要结论、关键数字、风险和决策项；项目组还需要过程、负责人、截止日期和下一步。除“专业、简洁”这类抽象形容词外，应给出页数、字数、标题层级、必含字段、可参考样例和禁止表达。制作 PPT 时再补充汇报时长、投屏可读性和每页一个核心观点。这样结果是否合用可以被观察，而不是依赖主观感觉，修改时也能准确指出应调整的页面或字段。先定读者，再定内容和形式。

14. 复杂任务为什么要先要计划？

答：Plan 更适合多步骤、跨系统和重要文件任务，原因是计划阶段能在真实操作发生前暴露问题。一个可检查的计划应列出每一步使用什么输入、调用什么能力、产生什么中间产物、在哪里需要人确认，以及失败时怎样停下。比如 Word 转 PPT，可以先确认材料范围和页面大纲，再生成可编辑文件；多 Agent 视频任务则必须先确认 Brief，再进入分镜和素材阶段。计划不是为了展示思考过程，而是把高成本返工和不可逆动作挡在执行之前。

15. 多步任务应该怎样拆？

答：拆分原则不是平均切成几段，而是让每一步都能独立验收并形成清楚的交接物。可以采用这个顺序：先确认任务简报和资料口径，再读取与清洗材料，随后分析并形成结构，接着生成文档、表格或 PPT，最后核验事实和版式。上下游角色只读取已经确认的产物，不靠口头转述关键内容。可以并行的步骤要写明汇合点，必须串行的步骤不能提前启动。这样一旦结果出错，可以追到具体阶段，而不是整项任务推倒重来，也能减少多角色之间的重复劳动。

16. 什么是验收标准？

答：验收标准是把“看起来不错”改写成可观察条件。例如，会议纪要要求每项结论能在原文找到依据，负责人和日期缺失时必须标待确认；文件整理要求清单数量与实际文件一致，确认前不得移动；PPT 要求数字与源文一致、页数受限、投影状态可读。好的验收同时覆盖内容、格式、路径和边界，并说明谁来确认。若没有验收，模型只能按自己的偏好交差，用户也很难指出应改哪一处、改到什么程度才算完成，更无法稳定复用同一流程和标准。验收要在实际文件中完成。

17. 输入资料很多时该怎么交代？

答：需要注意，上下文越长不一定越好，无关材料会淹没关键要求，旧规则也可能与新规则冲突。处理大量资料时，先建立清单，标明文件类型、日期、版本、用途和优先级；再区分唯一事实来源、仅供结构参考的旧稿，以及明确不得使用的过期材料。可以先让 WorkBuddy 输出目录和读取摘要，确认范围后再分析。若多份资料口径冲突，要求它列出差异和来源，暂停等待负责人裁决，而不是自行挑一份写进最终稿，裁决结果也要单独留痕。整理范围应在执行前确认。

18. 什么时候应该让 WorkBuddy 先追问？

答：应把“关键输入缺失、目标冲突、权限不足、成本超预算、动作不可逆、结果无法验收”列为应暂停的条件。遇到这些情况，应在任务中明确写“先列出缺失项和需要确认的问题，未确认前不要生成正式稿或执行写入”。例如会议记录没有负责人和截止日期，就标待确认；自动化没有 owner 和停用方式，就先不上线；外部发布目标不明，就只生成草稿。会停下来询问并不是能力弱，而是一个可靠执行系统必须具备的边界，也方便后续审计。暂停后还应说明怎样恢复。

19. 语音下任务要注意什么？

答：语音适合快速召唤任务，但它先要经过语音转文字，人名、日期、路径、金额和专业术语都有误识别风险。下达低风险查询时可以直接使用；涉及本地文件、外发或覆盖时，应先让 WorkBuddy 复述目标文件、操作动作和输出位置，确认无误再执行。长任务最好把语音当作需求草稿，随后补充六要素任务卡。手机端还要确认当前是本机模式还是云端模式，避免以为正在处理电脑文件，实际运行环境却不同。发送前最好再检查一次转写文字和目标路径。

20. 如何让修改意见更有效？

答：二次修改时，不要重新描述整项任务，而要说明“差异”。有效反馈应包含对象、问题和目标，例如“保留第三节结构，只把第二段压缩到八十字；数字不得改变；新增的结论必须有来源”。对表格可以指出哪一列口径错，对 PPT 可以指出哪一页信息溢出，对会议纪要可以指出哪条待办缺负责人。修改后只复核受影响部分及其上下游，不要每次都让模型整稿重写，否则容易把已经通过验收的内容再次改坏，并扩大不必要的返工范围。具体反馈也应保留原有正确内容。

A.3 文件资料与成果交付（21—30）

21. 引用文件后为什么还要说明用途？

答：“输入”和“动作”必须分开说明：引用文件只告诉 WorkBuddy 材料在哪里，并没有说明你要它怎样处理。同一份合同可以做摘要、版本对比、风险提取或格式转换，结果完全不同。引用后应写清处理范围、动作、唯一事实来源和输出，例如“只比较两个版本的责任条款，列出新增、删除和修改，不评价法律效力”。文件很多时先要读取清单和摘要，确认它识别了正确版本。若输入只是参考风格，也要明说不可把其中事实带入新稿。处理目标越明确，文件引用才越有价值。

22. WorkBuddy 没读对文件时先查什么？

答：可以按这个顺序排查：确认工作目录是否正确、系统是否授予对应目录权限、文件是否被其他程序锁定，再检查格式和版本。不要在重要原件上反复试错，可以先用空白文本文件验证读写是否正常。正式处理前，让 WorkBuddy 返回它实际看到的文件名、数量、修改时间和一小段可识别内容，或生成 inventory 清单；这些信息与目录一致后再继续。若读取结果不对，先修路径和权限，不要靠重复改 Prompt 掩盖基础连接问题。排查后保留测试结果和错误截图。

23. 大文件应该一次性全部交给它吗？

答：把所有文件和历史对话一次塞入任务，可能造成旧要求冲突、关键资料被淹没、等待与成本上升，并引用过期版本。更稳妥的做法是先建立文件目录和章节地图，按本次目标限定页码、时间段或字段，再分批生成可核查的中间摘要。跨文件分析时，将已确认事实、当前规则和决策记录单独落盘，下一阶段只携带必要内容。拆分后还要设置汇总步骤，检查各批次是否遗漏、重复或口径不同，避免分段正确但合并失真，最终摘要仍需回到原文抽查。每一批都要保留来源和编号。

24. 多个版本的文件怎样避免用错？

答：版本混乱本质上是资料边界不清。先确定“主版本”，再连接工具或交给 Agent 使用。文件名至少包含主题、状态、版本或日期，例如“制度_批准版_v2”，并将原件、工作副本、待审核稿和最终成品放在不同目录。任务说明中明确哪份是唯一事实口径，哪些旧稿只参考结构。遇到两份都自称最终版时，不要让 WorkBuddy 自行选择，应先生成差异表，记录来源、时间和冲突字段，由负责人裁决后再继续。裁决结果还要记录日期和负责人。

25. 搜网页时怎样减少不可靠信息？

答：调研时应把官方站点、官方发布、产品文档和实际测试放在优先位置，视频口播和二手文章更适合作为线索。网页调研任务应要求保存标题、链接、发布或更新时间、关键证据和检索日期；对产品功能、政策、价格、数据和人物信息使用至少两个独立来源交叉验证。若只有单一口播，或页面没有日期，就把结论标为待核查，不要改写成确定事实。最终稿应区分资料直接支持的事实、基于事实的推断，以及作者自己的建议，并给待核查项保留明确位置。

26. 为什么网页调研要限定时间范围？

答：网页和视频往往发布于不同时间，同一功能在早期演示、后续更新和第三方口播中可能并不一致。检索任务应写明“截至某日”，并要求每条证据同时保留内容发生时间、页面发布时间和本次访问时间。对没有日期或只说“最新”的材料降低权重，对旧教程里的界面路径采用版本中性表述，并在实际客户端再次核验。若要比较趋势，应把不同时间点分开呈现，不能把旧价格、新功能和当前政策拼成一张看似同步的结论表，同时注明具体版本差异。重要结论应在发布前再次核验。

27. Markdown、Word 和 PDF 应该怎么选？

答：不同文件格式适合不同使用阶段：Markdown 结构清楚、便于快速改稿和版本比较，适合作为内容底稿；Word 或在线文档适合正式排版、批注、协作和修订；PDF 更适合锁定版式后阅读、归档和分发。稳妥流程是先在可编辑格式中核对事实与结构，再生成 Word 检查标题、表格和编号，最后导出 PDF 做逐页视觉复核。若任务还会继续自动处理，就同时保留机器可读的源文件，不要只留下难以修改的 PDF 或截图，并记录各格式的生成关系。

28. 生成文件后最先检查什么？

答：生成完成后，先在结果区查看文件、变更和预览，但预览通过不代表正式文件一定可用。第一层检查文件名、路径、格式、大小和能否在本地应用中打开；第二层核对章节是否齐全、数字是否回到源文件、表格公式能否刷新、PPT 是否可编辑；第三层逐页检查分页、字体、图片、链接和投影可读性。分享前再扫描客户信息、账号、内部路径和未公开内容。若导出过程中出现缺页或版式损坏，应回到可编辑源文件修正，而不是只改对话里的文字。检查结果最好形成简短验收记录。

29. 为什么要区分原始资料、过程稿和成品？

答：应按任务隔离工作目录，因为文件夹级授权会直接决定 Agent 可以读写什么。建议至少分为 input 或 original、working 和 output：原始资料只读，过程区保存清洗结果、计划与草稿，输出区只放通过验收的交付物。这样既能防止覆盖原件，也能识别哪个文件仍待确认。若要整理或移动文件，先输出 inventory 和 proposed-actions，人工核对数量、重名与目标路径后再执行。长期项目还应把旧成品归档，并保留版本和变更记录。

30. 怎样让结果便于继续修改？

答：交付物应当能够继续编辑和使用，而不是只看起来完成。文档使用稳定标题层级，表格保留字段、单位、公式和数据来源，PPT 同时提供逐页内容清单与可编辑文件；无法确认的事实单列，不要藏在正文中。还应规定命名、保存路径和版本号，方便下一轮只修改受影响部分。图片或 PDF 可以作为阅读版，但不能取代源文件。结构化结果既便于人工对照，也能被后续 Skill、自动化或其他 Agent 安全读取。这样才能继续修改、比较和自动处理。

A.4 知识库与资料管理（31—40）

31. 知识库解决的核心问题是什么？

答：可以把知识库理解成“随时可查的资料室”：它保存制度、产品、案例、SOP 等可检索材料，解决 WorkBuddy 依据什么回答，而不是让模型永久记住全部内容。实际回答通常还经过检索增强过程，系统先找到相关片段，再交给模型生成。因此，接上知识库并不等于自动正确；资料来源、分段方式、元数据、版本和检索是否命中都会影响结果。使用时应要求给出引用位置，并允许回答“资料中无法确认”，这样才能把流畅表达和真实依据分开。

32. 哪些资料适合放进知识库？

答：适合进入长期知识库的，是会被反复查询且来源清楚的内容，如批准版制度、产品手册、标准问答、项目决策、会议结论和经核验的学习笔记。每份资料最好带标题、来源、日期、版本、适用范围和维护人。临时下载、未确认草稿、聊天碎片、重复文件和营销口播不要直接混入主库，可先放收集区等待整理。敏感资料还要先判断是否允许进入该系统、是否需要脱敏和分权。知识库不是大杂物箱，资料越可追溯，回答才越能被信任，后续更新和纠错也更容易定位。

33. 知识库是不是文件越多越好？

答：不是。需要注意，资料差、切分不合理、检索偏差，回答就会跟着失真。大量重复文件会让同一观点反复出现，过期制度可能压过新口径，互相冲突的版本还会诱导模型随意选择。更好的做法是先确定主版本，给材料补齐来源、日期、标签和适用范围；重复项合并，旧版移入归档并标注失效原因。更新后用一组固定测试问题验证事实命中、跨文件比较、冲突识别和“无答案”处理，而不是只看知识库文件数量是否增长或导入速度是否很快。资料治理比盲目扩充数量更重要。

34. 本地文件夹也能当知识库吗？

答：可以，但本地文件夹只有经过组织后才称得上可用资料系统。应按项目隔离目录，先确定主版本，再统一文件名、分类、标签和状态；原件、过程稿、已批准内容和归档旧版分别存放。WorkBuddy 是否能读取，取决于当前任务的目录授权、文件格式和锁定状态，不能因为文件在电脑上就假定模型看得到。长期资料还应备份，并为重要规则记录来源、日期和负责人。若使用 Obsidian，本质上仍要维护 Markdown、链接和版本，而不是只安装工具。

35. 知识库回答太泛怎么办？

答：先区分“库里没有”“没有检索到”和“任务问得太宽”三种情况。查看相关资料是否真实存在、格式能否解析、版本是否有效，再让 WorkBuddy 列出本次命中的文件和引用片段。把“介绍一下产品”改成限定对象、时间、字段和输出结构的问题，例如只根据批准版手册比较三项能力。若引用仍与问题无关，就需要调整资料分段、标题、标签或关键词，而不是不断换模型。资料本身没有答案时，应明确返回无法确认并列出缺口，不能用通用知识补齐内部事实。

36. 多份资料互相冲突怎么办？

答：处理资料冲突时，冲突应显式暴露而不是由模型静默裁决。先让 WorkBuddy 按文件名、来源、发布日期、版本、适用范围和冲突字段生成对照表，并引用各自原文；再判断是否存在明确的批准版、更新通知或维护负责人。没有权威依据时，将冲突项标记为待确认，暂停下游写作或自动化。负责人作出选择后，把采用口径、理由、日期和失效版本写入决策记录，同时更新知识库。否则同一冲突会在后续每次检索中重复出现，并持续污染报告和自动化结果。

37. 知识库和项目有什么区别？

答：知识库更像可检索的资料室，项目则是围绕目标组织上下文、任务状态和产物的工作空间。知识库保存可长期复用的制度、产品和案例；项目保存本次目标、已确认事实、决策记录、工作副本、进度与成品。一个项目可以读取多个知识库，但不能把项目里的临时猜测自动写回长期资料。项目结束时，应先区分哪些结论已批准、哪些只是过程信息，再选择性沉淀。这样能避免短期任务污染长期知识，也能让下一次续作从已确认状态开始。两者配合使用，但承担不同职责。

38. ima 和 Obsidian 该怎么选？

答：选择工具时，不必比较谁更强，而要先确定主版本和资料寿命。ima 更适合作为移动收集、快速问答和阶段性生成入口；Obsidian 强调本地 Markdown、链接、标签和可维护的长期 Wiki。选择时应看资料是否需要多人协作、是否必须本地保存、更新频率和隐私边界。两者可以组合，但要指定哪一处是主版本，避免同一笔记在多个工具里分别修改。WorkBuddy 负责把这些资料转成任务结果，不能替代命名、版本、备份和维护责任。

39. 团队知识库最容易忽略什么？

答：最容易忽略的不是导入，而是持续治理。团队库应规定谁可以新增、修改和废止资料，文件如何命名，哪些成员能看敏感内容，何时复核有效期，以及冲突由谁裁决。重要资料附上来源、版本、适用部门、负责人和下次复核日期；旧版不能简单删除，也不能继续参与默认检索，应归档并标明失效。每次规则更新后还要重跑固定测试问题，确认新口径能够被检索。没有 owner 的知识库会迅速变成旧制度、新制度和重复文件并存的噪声仓库，越用越难建立信任。

40. 如何验收知识库是否可用？

答：验收时应准备多种类型的问题，而不是只问一个容易命中的事实。测试集至少包含：单文件事实、跨文件组合、同名版本冲突、边界条件、资料中没有答案，以及包含过期材料的干扰题。每个回答检查是否引用到正确文件和片段，能否区分原文事实、推断和建议，缺失时是否明确停下。更新资料、分段或检索设置后重复同一组测试，并记录通过率和失败原因。回答写得长、语气很肯定，都不能替代可追溯证据和持续稳定的测试结果。测试题也要随业务变化更新。

A.5 Skill、连接器与 MCP（41—50）

41. Skill 到底是什么？

答：Skill 是一组可复用的说明、脚本、参考资料、模板和资源，用来告诉 Agent 某类任务怎样做、调用什么工具、交付什么格式。最小结构通常有 SKILL.md，复杂 Skill 还会带 references、scripts、templates 和 tests。它的价值不是让模型参数变强，而是把容易遗漏的步骤、业务口径和检查方法固定下来，并按需加载，减少每次复制长 Prompt。可以把 Prompt 理解为本次任务，把 Skill 理解为一类任务的稳定做法；装上后仍需权限审查和结果验收。

42. 知识库和 Skill 有什么区别？

答：可以用两个问题区分它们：知识库回答“依据什么”，Skill 回答“怎样稳定地做”。产品手册、制度和历史案例适合放知识库；从这些资料检索事实、核对版本、生成销售问答并做格式检查的步骤，才属于 Skill。Skill 可以在运行时读取知识库，也可以包含少量专用参考文件，但不应把大量事实硬塞进操作说明。验收时也要分开：回答事实错，先查资料和检索；步骤漏了、格式漂移，再查 Skill。把两者混在一起，会让更新事实和更新流程都变得困难。

43. 什么时候值得自定义 Skill？

答：应从 Prompt、任务卡、SOP 到 Skill 逐级固化，而不是第一次成功就封装。适合做 Skill 的任务通常反复出现，输入类型相对稳定，步骤和异常分支已经明确，输出格式可检查，并且团队确实需要复用。先用真实样本手动跑三五次，记录哪里需要追问、哪些动作要确认、失败时怎样降级；再把稳定规则写入 SKILL.md，把业务资料、脚本、模板和测试分别归位。仍在频繁改变的流程应继续使用任务卡，否则 Skill 只会把未经验证的错误放大。

44. 一个可用的 Skill 至少要写什么？

答：一个可用的 Skill 至少要有准确的 name 和 description，让 Agent 知道何时触发；正文要写明输入要求、步骤、调用工具、输出格式、权限边界、停止条件和异常处理。复杂任务把业务口径放 references，可重复计算放 scripts，固定版式放 templates，正常、缺字段、冲突和失败样例放 tests。尤其要说明资料缺失时返回什么、哪些动作必须人工确认、怎样停用或回退。只有“帮我专业地完成任务”这类泛化说明，不足以形成稳定能力，也无法做版本和回归测试。

45. Skill 调用失败时怎么排查？

答：可按“触发—输入—依赖—执行—输出”顺序排查。先确认 Skill 名称、description 和启用状态是否让 Agent 正确选择；再检查本次输入是否满足要求，引用文件能否读取；随后确认脚本、模板、网络、连接器和凭证是否可用。查看日志判断是未触发、执行报错还是产物验收失败。可以用同一目标改成自然语言最小任务：基础能力能完成而 Skill 失败，问题多半在 Skill；两者都失败，则先查权限或工具。第三方 Skill 还要检查来源、更新状态和停用方法。

46. 连接器的作用是什么？

答：连接器是 WorkBuddy 与外部工具、数据和消息渠道之间的入口。连接成功后，Agent 才可能在授权范围内读取文档、查询系统、创建记录或回传结果；具体能做什么取决于连接器暴露的工具、当前账号身份和权限。它解决的是“连到哪里、可调用什么”，不会自动规定业务步骤，也不会保证结果正确。加载后应先执行只读测试，确认账号、资源范围和返回内容；涉及写入时再逐项扩大权限。连接成功的图标也不能替代一次真实的端到端验证。

47. 连接器和 Skill 如何配合？

答：连接器负责接触外部世界，Skill 负责把任务按稳定方法完成。一个典型链路是：连接器获取会议、日历或协作文档，Skill 按规则生成结论、待办、负责人和待确认项，随后再通过连接器写入任务系统或发送通知。两层应分别验收：读取不到资料时查账号、授权、网络和资源范围；资料已返回但结构错误时查 Skill 的字段与步骤。写入外部系统前增加人工确认，并记录创建结果的 ID，避免重试造成重复任务或重复消息。这样故障定位不会把“连不上”和“做不好”混为一谈。

48. MCP 可以怎样理解？

答：可以把 MCP 理解为 AI 工具生态的通用接口：工具方按统一方式暴露 Tools、Resources 或 Prompts，支持 MCP 的客户端就能发现和调用它们，减少为每个 AI 与每个系统重复适配。它解决的是“怎样连接和描述能力”，不自动解决合规、密钥保管、最小权限、结果正确性和生产写入风险。小白不必先研究协议实现，但每次接入仍要问清几件事：使用谁的身份、能读什么、能改什么、数据到哪里、失败如何停止和回退。敏感连接更适合按项目隔离。

49. 外部连接授权越多越好吗？

答：需要注意，Agent 的风险不仅是“说错”，还可能真实修改文件、发送消息或写入系统，因此权限越大不等于效率越高。接入时从最小范围开始：使用专用或低权限账号，只授权当前项目需要的目录与资源，能完成验证就先用只读；写入、删除、发布、付款、群发和生产数据库操作设置人工确认。还要了解数据是否外发、凭证保存在何处、怎样撤销授权。任务结束后关闭不再使用的连接，定期检查授权有效期和访问日志，避免长期遗留高权限入口。

50. 连接器失效先检查哪些地方？

答：先确认失败发生在“连接、读取、处理还是写回”哪一层。连接层检查账号是否正确、授权或凭证是否过期、网络与回调是否可达；读取层确认目标文档、群或数据库对该账号可见；写回层再看机器人、应用和成员范围是否允许操作。团队平台还可能需要管理员发布应用或开通事件权限。查看错误码和日志，401/403 一类认证问题不要盲目自动重试，应转人工修复配置；临时超时才适合有限重试。修正后先跑只读最小测试，再恢复正式任务，并观察一次完整的端到端结果。

A.6 助理、项目与多 Agent 协作（51—60）

51. 助理适合承担什么工作？

答：助理更适合承接长期、重复且职责边界清楚的工作，如接收移动端任务、整理日报、追踪会议事项或定期收集某一主题。配置时不能只写一个人格名称，应明确服务对象、主职责、可用工作区、常用资料、默认输出格式和必须暂停的情况。先用只读任务测试它是否连接到正确账号与目录，再逐步开放写入。若一个助理同时负责财务、内容、行政和开发，资料与规则容易串用；更稳妥的做法是按职责拆分，并为每个助理保留独立的授权和验收标准。职责清楚后，助理才容易长期稳定。

52. 助理和专家有什么区别？

答：专家可以理解为带有特定领域方法、判断标准和工具组合的角色，助理则更偏向长期入口、日常职责和稳定习惯。助理可以持续接收任务、调用资料并回传状态；专家适合对某个专业子任务给出结构化分析或审查意见。两者都不天然拥有现实资质，区别也不是称呼更高级。配置专家时应说明输入、专业边界、证据要求和交付格式；配置助理时更重视工作区、渠道、默认习惯和升级路径。高风险结论无论来自助理还是专家，都需要有权限的人复核。角色名称不能替代专业审核。

53. 为什么助理最好只设一个主职责？

答：多 Agent 协作要强调角色契约和最小必要上下文：每个角色只接收完成本职工作需要的资料，并按固定格式交接。助理同样适用这个原则。职责太多时，不同资料权限、语气、工具和验收标准会互相干扰，甚至把一个项目的规则带到另一个项目。先用一个清晰职责跑通，例如只

负责会议后续或内容选题；记录它常见的输入、产物和异常，再决定是否扩展。需要跨专业协作时，由主理人拆成多个角色，并通过已确认的共享产物协作，而不是把所有能力都塞进一个全能助理。

54. 什么时候应该建立项目？

答：当工作会跨越多次对话、持续使用同一批资料、反复产生中间产物和决策时，就应该从临时任务升级为项目。不要依赖无限延长的聊天记录，而是把当前规则、已确认事实、决策记录、任务状态和文件路径落盘。项目开始时建立独立目录，写清目标、范围、主版本、资料入口和最终交付；每个阶段结束后保存状态摘要。这样即使任务中断、模型切换或换人接手，也能从已确认产物继续，而不会重新猜测历史。临时闲聊和无关资料不要混进项目上下文。

55. 项目里应该保留哪些信息？

答：项目至少应保留六类信息：目标和边界、资料及主版本清单、角色分工、已确认决策、当前状态与下一步、各阶段产物路径。重要交接不应只存在角色之间的对话里，而要写入共享文件；下游只读取上游已经确认的 Brief、数据表或分镜等产物。每次规则变化时记录日期、负责人和影响范围，旧决定标记失效而不是悄悄覆盖。还要保存失败原因和待确认项，便于断点续跑。项目摘要的作用是恢复状态，不是替代原始证据。共享文件还应注明版本与更新时间。

56. 多 Agent 适合什么任务？

答：先判断任务是否真的值得拆：要有不同专业工序、部分工作可以并行、各角色有明确交付格式，并且最终有人能统一验收。例如产品宣传片可拆成 Brief、分镜、素材、剪辑和配乐，但每步都有前后依赖。若只是改一段文案或整理一张表，单 Agent 或固定 Workflow 更可控。多 Agent 会增加上下文传递、等待、成本和失败传播，不能把角色数量当作质量指标。决定使用前先画出串行步骤、并行步骤、汇合点和三个必须由人确认的节点。

57. 多 Agent 怎样避免结果互相矛盾？

答：专家团协作需要“共享产物层”和“主理人”两个关键机制。所有角色不要各自维护一份产品事实，而应读取同一个已确认 Brief、数据表或规则文件；角色契约写清输入、输出、工具权限和完成条件。可以并行的工作在汇合点统一检查，必须串行的工作等待上游确认。主理人负责分发最小必要上下文、检查交接格式、处理冲突和合并最终产物。合并时要核对术语、事实来源、重复和遗漏，不能简单拼接。若上游事实有误，应先修共享产物，再重跑受影响的下游角色。

58. 专家给出的结论可以直接采用吗？

答：不能。模型生成能力不等于现实责任：专家只是预置或自定义的专业工作角色，可能组合特定 Prompt、Skill、工具和模型，但名称本身不构成执业资格，也不保证事实无误。使用时要求它列出材料来源、分析假设、适用范围、相反证据和无法确认项；涉及法律、财务、医疗、投资或企业制度时，再由具备资质或授权的人审核。专业角色的价值是让分析框架更完整、遗漏更少，而不是替用户作最终决定。若资料不足，它应暂停并请求补充，而不是用“专家口吻”填空。

59. 长任务中断后怎样继续？

答：可以用状态文件记录批次、当前状态、已完成步骤、来源状态、最后错误和更新时间；普通长任务也可采用同样思路。恢复前先读取最近的状态摘要和已确认产物，核对输入文件、规则、权限或外部数据是否已变化。续做指令明确“从哪个版本、哪个步骤继续”，并跳过已经成功的阶段。若中断发生在交付或推送阶段，应复用已生成结果，不要重新抓取和分析。恢复后检查

是否产生重复文件、重复消息或覆盖风险，再更新状态记录。每次恢复都要确认断点是否仍然有效。

60. 为什么项目要定期复盘？

答：复盘是从“一次能用”走向“稳定复用”的入口。项目结束后，记录哪些输入最常缺失、哪个步骤反复返工、什么验收条件真正发现了问题、哪些异常需要人工处理，以及本次时间和成本。把有效 Prompt 抽成任务卡，把固定步骤写成 SOP，把脚本和模板归入 Skill；只有手动多次稳定运行、触发与 owner 明确后，才升级为自动化。复盘还要删除只对单一案例有效的细节，保留可替换字段，并用另一组真实样本验证，避免把偶然成功误当成通用方法。

A.7 自动化与远程任务（61—70）

61. 自动模式和自动化有什么区别？

答：默认模式、Ask 和 Plan 是当前任务的工作方式。Ask 偏理解和查看，Plan 适合先列出步骤与边界；需要执行文件操作时，还应核对任务权限。本书第 1 章所示界面没有 Craft 入口，不要照旧教程寻找。自动化则把已经稳定的 Prompt、数据源、触发时间、输出和推送配置保存下来，以后按条件重复运行。两者控制的层级不同：当前模式决定这一次如何推进，自动化决定未来什么时候再次触发。一个任务能手动成功一次，并不代表已经适合定时运行；还要补上 owner、幂等、日志、告警、停用方法和异常分支，才可能成为可靠自动化。

62. 哪些任务适合自动化？

答：自动化前要跨过三个门槛：同一 Prompt 至少手动运行三次，输出质量和格式基本稳定；触发条件、输入来源和验收标准清楚；有 owner、告警和停用方法。适合的例子包括固定数据源的每日资讯、周期报表或重复文件处理。上线前还要演练数据源超时、结果为空、推送失败和重复触发，确认系统能降级而不是伪装成功。若 Prompt 和数据源还在频繁调整，先继续手动运行。自动化的目标不是完全无人，而是正常路径少打扰人，异常时能找到正确的人。

63. 哪些任务不适合自动化？

答：目标和路径经常变化、输入无法稳定获取、结果难以量化验收、依赖品牌或专业判断的任务，不适合直接全自动。不可逆动作必须设置人工确认点，包括对外发布、任务系统写入、删除、付款和生产环境修改。可以采用半自动方案：自动收集资料、去重、生成候选清单或草稿，然后由人确认方向和最终动作。若任务每次都要临场决定下一步，Agent 可能比固定 Workflow 合适，但仍要设置停止条件、预算和日志。自动化不是把不确定性藏起来，而是把稳定部分机械化。

64. 自动化任务至少要定义哪些要素？

答：一份完整的自动化任务配置应包含：任务名称、触发方式、完整 Prompt、数据源、质量门禁、输出结构、推送目标、幂等控制、重试策略、告警接收人、owner 和停用方法。除此之外，还要说明空输入、部分数据源失败、权限失效和成本超预算时的状态。每次运行生成唯一批次 ID，并记录哪些步骤完成、结果保存在哪里。这样才能回答“这次是否真的成功”“重跑会不会重复发送”“失败后谁处理”。只写“每天九点帮我整理一下”只能创建一个脆弱定时任务，不能构成可靠工作流。

65. 为什么自动化必须保留日志？

答：每次运行应记录批次 ID、触发方式、各数据源状态与耗时、输入和过滤数量、推送目标与结果、总耗时、错误信息以及 Token 或 API 调用成本。日志的价值不是堆满正文，而是让人

能复现状态、定位失败和判断是否需要重跑。推送失败时，日志应证明内容已经生成，只需从 delivering 继续；认证失效时，则应转人工而不是盲目重试。日志还支持长期查看按时触发率、一次成功率、来源可用率和成本趋势。没有这些记录，所谓“自动完成”无法审计，也无法改进。

66. 怎样防止自动化重复执行？

答：可以用“幂等”解决重复触发：每次运行生成唯一批次 ID，例如任务名加日期，关键步骤完成后写入状态。再次收到同一批次时，先检查抓取、分析、文件生成和推送分别是否成功；已经完成的步骤直接复用，不重新执行。对消息推送还应保存 message ID，对文件输出检查目标是否存在和内容是否一致。同一批次已完整成功，就返回已有结果；只在中途失败，才从断点继续。这样可避免手动触发与定时触发撞车，也能防止网络重试造成重复通知或重复写入。

67. 远程给 WorkBuddy 下任务要注意什么？

答：小程序通常要先区分本机模式和云端模式：本机模式在已连接电脑执行，依赖电脑在线，适合本地文件和工作区；云端模式在隔离环境运行，不依赖电脑在线，更适合调研和临时分析。远程下任务前先确认当前模式、目标电脑、账号和授权目录，再用只读指令验证链路。任务中写明输入路径、输出位置和手机端需要收到的摘要。删除、覆盖、外发等高风险动作不要仅凭模糊语音执行，应先返回计划和目标清单，由手机端确认后再继续，并保留确认记录。

68. 远程任务没有响应怎么排查？

答：先判断你使用的是本机还是云端模式。本机模式下检查电脑是否开机、未休眠、网络正常，WorkBuddy 桌面端是否登录且连接到正确账号；随后查看小程序或 IM 助理是否仍显示绑定，二维码或凭证是否过期，消息是否进入队列。若使用飞书、钉钉等企业应用，还要核对机器人发布状态、事件订阅和成员权限。保留发送时间、指令原文、目标电脑、回执、错误提示和桌面端日志。先用一条只读测试恢复端到端链路，再重发正式任务，避免重复执行。

69. 自动化遇到空输入应该怎么办？

答：自动化状态机应把“数据源可达”和“内容有效”分成两道门禁。空输入可能是当天确实无数据，也可能是权限、超时或来源异常，不能统一当成功。任务应先记录各来源状态：来源正常但无符合条件内容，可以输出“本轮无有效条目”；来源全部失败则进入 blocked 并告警。不要沿用上一批正文冒充新结果，也不要用空白文件覆盖旧产物。保留批次 ID、触发时间和失败原因，次日可正常重新触发；若需要人工补数据，告警中写清入口和处理建议。

70. 自动任务什么时候必须暂停等待确认？

答：至少六类情况要暂停：关键输入缺失、目标或资料冲突、权限不足、成本超过预算、动作不可逆、结果无法验收。具体到自动任务，还包括同名文件将被覆盖、认证失效、对外发布和任务系统批量写入。暂停时不要只报“失败”，应返回批次、已完成步骤、具体影响、待确认选项和建议动作。收到确认后，把确认人、时间、选择和恢复步骤写入状态，再从断点继续。一个可靠自动化不是永不停顿，而是知道何时必须把决策权交回给人。暂停不是失败，而是风险控制的一部分。

A.8 办公与内容应用（71—80）

71. 怎样让会议纪要更可执行？

答：会议纪要可以分为会议结论、待办事项、负责人、截止日期和待确认问题，并要求每项结论能在原文找到依据。实际操作时先保存原始录音、转写或会议记录，再让 WorkBuddy 按字

段提取；没有明确负责人或日期的内容写“待确认”，不要根据发言人或常识补全。会后先让参会者确认纪要，再将已确认待办写入任务系统。需要特别注意，不能把模型提取的待办静默写入外部系统，否则一次误识别就会变成真实任务和错误通知。确认后的纪要再作为后续执行依据。

72. 会议纪要为什么不能只做摘要？

答：摘要主要压缩讨论内容，而会议纪要要推动后续工作。会议闭环应从“会前定义要做出什么决定”开始，会后不仅整理讨论，还要区分正式决议、建议、未决问题和可执行待办。待办应包含任务、负责人、截止时间、交付物和状态；无法从转写确认的字段留空并标记。纪要确认后再发送通知、创建任务并跟踪完成情况。验收时随机抽取几条结论回到原始转写定位，检查是否把意见误写成决议、把模糊时间补成具体日期，或遗漏反对意见和待确认事项。

73. 周报怎样避免写成流水账？

答：先明确周报给谁看、帮助对方做什么判断，再决定结构。输入可包括日报、任务系统记录、会议纪要和项目数据，但要注明哪份是事实来源。让 WorkBuddy 按项目合并同类动作，区分已完成、进行中、阻塞、风险和下周计划，并保留可核对的数字；不要把“开会、沟通、跟进”当作成果。管理层版本突出结论、偏差和需要支持，执行版本再保留责任人与日期。生成后回查原记录，删除重复事项，对没有证据的“显著提升”等评价降级为待确认。

74. 日报怎样快速变成周报？

答：日报能否转成周报，取决于日常记录是否结构化。日报至少应包含当天完成、进行中、阻塞、下一步和关键数据，并给每项标项目或主题。汇总时让 WorkBuddy 按项目而不是按日期聚合，合并连续多天的同一事项，保留最终进展与未解决问题，删除重复的过程描述。阶段成果要能回到日报、任务记录或数据表核实；缺少基线的数据不能直接写成增长结论。最后根据周报受众调整信息密度，并把需要领导决策或跨部门支持的事项单独列出。周报最终仍需负责人核对。

75. 用 WorkBuddy 做数据分析先检查什么？

答：处理 Excel 时，先读取表头和样例行，检查字段类型、日期、单位、空值、重复、异常值和公式，再确认业务指标的定义、时间范围和唯一数据口径。清洗前保留原始表，只在工作副本操作，并输出数据质量报告和异常清单。若关键字段缺失或多个表口径冲突，应先暂停请业务负责人裁决，而不是直接补零或删除。质量门通过后再做汇总、图表和结论；每个指标保留计算方式与源字段，确保表格公式可刷新。漂亮图表无法弥补错误数据。分析前还要确认数据是否完整。

76. 数据分析结果怎样区分事实和建议？

答：模型语言自信不等于证据充分。输出报告时可分成三栏：事实只写数据直接支持的变化，并标出指标、单位、时间范围和源表；推测说明可能原因、依据和仍需验证的假设；建议则写出动作、负责人、优先级和预期验证方式。不要把两个指标同时变化直接写成因果，也不要把缺失值处理后的结果伪装成原始事实。对重大异常返回明细行或计算过程，让人工抽样复算。这样读者能清楚知道哪些可以引用，哪些只是下一步调查方向。事实、推测和建议不能混写。

77. 做 PPT 应该先美化还是先定结构？

答：演示稿应该是“把材料变成叙事”，而不是给文字套模板。先明确汇报对象、使用场合、时长和希望对方作出的决定；再从事实材料提炼核心结论，生成背景、现状、问题、方案、计划和决策项等页面大纲，并让每页只承担一个观点。大纲通过后才选择图表、版式和视觉风格，同

时引用品牌规范。生成可编辑 PPT 后要检查数字与源文件一致、文字是否溢出、投屏字号是否可读、图片是否有来源。结构未定就美化，会把返工扩散到每一页。先定主线，再谈视觉表现。

78. 怎样用 WorkBuddy 写公众号文章？

答：可以把内容生产看成选题、素材、结构、初稿、发布和复盘的闭环。写作前说明账号定位、目标读者、本篇解决的问题、已有素材和希望读者采取的动作；先让 WorkBuddy 输出选题判断与大纲，确认后再逐段写。热点和案例要保留来源、日期与可引用范围，不能把视频口播当成已验证事实。成稿后检查标题与正文是否一致，删除夸张承诺和模板化套话，确认图片版权与隐私。发布仍由人确认，发布后的阅读、收藏和反馈再进入复盘库，指导下一篇。

79. 怎样让短视频脚本更容易拍？

答：短视频生产可拆成信息采集、内容策划、分镜、素材、制作、字幕与配乐等工序。任务不能只要一段口播，应同时规定目标受众、时长、核心信息、参考风格和已有素材，并输出镜头编号、画面内容、动作、口播、字幕重点、素材来源和预计秒数。先确认 Brief 和分镜，再生成素材与成片；素材生成失败时可交付分镜和缺口清单，不得伪装完成。对产品卖点、截图和引用做事实核验，音乐、图片和视频确认授权，最终由人判断节奏、品牌调性和发布风险。

80. 内容工作站应该沉淀哪些资产？

答：真正可复用的资产不只是成稿，还包括选题候选、原始素材及来源、标题实验、结构模板、发布版本、平台数据和复盘结论。每条素材记录采集日期、主题、版权或可用范围和关联作品；选题记录为什么采用或放弃；模板只保留稳定结构，不把旧事实带入新内容。已发布作品与草稿分开，复盘数据按平台和时间窗口统一口径。WorkBuddy 可以帮助去重、打标签和生成候选，但主版本、淘汰规则和复盘判断仍由人维护，避免素材库变成新的信息垃圾场。

A.9 权限安全与问题排查（81—90）

81. 文件权限应该怎么给？

答：工作目录既是效率设置也是安全边界；“完全访问”会让智能体读写授权目录之外的文件。实际授权应从按任务隔离的演练目录开始：先只读，再允许在 output 新建文件，确认无误后才考虑更高权限。input 目录保持只读，移动、覆盖、删除和外发前返回目标清单等待确认。连接器也采用同样原则，使用低权限账号和项目级范围，不把敏感生产连接设为全局。任务结束后撤销不再需要的授权，并检查实际文件变更和日志是否符合预期。权限应随着任务结束及时回收。

82. 哪些资料需要先脱敏？

答：高敏感内容包括个人身份和联系方式、客户与员工信息、合同及报价、财务与医疗数据、账号凭证、API Key、内部策略、源代码和未公开项目。处理前先依据组织制度判断是否允许进入当前工作区或外部系统；必要时用编号、区间、虚构样例或遮挡替代真实标识。脱敏不能只改正文，还要检查文件名、表格隐藏列、批注、图片、元数据、日志和导出版本。若任务需要保留身份映射，应单独加密保存并限制访问，不要和处理材料放在同一授权目录。

83. 为什么不应让 AI 直接清理整个磁盘？

答：需要注意，聊天模型说错可能只是误导，Agent 做错却会真实影响文件和系统。整盘清理范围过大，文件用途、系统依赖、符号链接和备份状态又难以从名称判断，不应直接交给自动模式。安全流程是先限定到明确目录，只读扫描并输出候选文件、大小、修改时间、判断依据和

风险；人工逐项确认后，优先移动到可恢复位置，而不是永久删除。执行前验证目标路径和数量，执行后生成变更记录。遇到锁定、路径异常或权限扩大应停止，不要自动换更强删除方式继续。

84. WorkBuddy 输出事实错误时怎么处理？

答：应把事实提取与建议生成分开，因此发现错误后先追溯链路：输入文件是否正确，知识库是否检索到旧版，网页来源是否过期，Prompt 是否允许猜测，还是计算或模型推断出错。要求 WorkBuddy 为错误结论列出引用位置、计算过程和假设；修正源数据或主版本后，只重跑受影响步骤，并检查下游表格、PPT 和结论是否继承了错误。高影响数字由人工回到原文件复算。不要只在最终段落改一个数字，否则摘要、图表和建议可能仍基于旧值，形成表面修正。

85. 输出格式总是不对怎么办？

答：输出格式要写成可检查的结构，而不是“更专业、更好看”。明确文件类型、标题层级、字段名称与顺序、字数或页数、命名规则和保存位置；表格说明单位与公式，PPT 说明页面结构和投屏要求。提供一个合格样例时，要注明只模仿结构还是也参考内容，避免旧事实被复制。先生成一页或一小段确认格式，再批量处理。若调用 Skill，检查模板和当前要求是否冲突；修改时指出具体差异，只重做相关部分，并在最终文件中验证，不以对话预览代替交付检查。

86. 长任务跑偏时应该继续还是重开？

答：发现跑偏后先暂停，取回当前计划、工具调用、中间产物和状态记录，与最初目标、资料范围和验收标准逐项比较，找出最早偏离点。若上游 Brief 或数据口径错误，先修共享产物，再重跑所有受影响下游；若只是某一页格式问题，只局部返工。目标、权限或主版本已经改变时，最好新建任务，明确引用哪些已确认结果，避免旧上下文继续干扰。恢复前再设置停止条件和人工确认点。暂停后还要判断已经执行的动作是否需要回滚，并记录跑偏原因，避免下一次任务再次走上同一路径。

87. 怎样排查“内容太泛”？

答：可以按五层排查。第一层看目标和受众是否具体，“介绍一下”通常只能得到通用内容；第二层检查输入材料是否存在、是否为主版本；第三层让它列出实际引用片段，判断知识库检索是否命中；第四层检查 Skill 的方法和输出模板是否适合本任务；第五层才考虑模型与上下文。上下文过长也会让关键内容被淹没，换更强模型未必解决。修正后先要求一段带来源的小样，确认信息密度和口径，再继续整篇。资料没有答案时，应承认缺失而不是用常识补齐。

88. 怎样控制使用成本？

答：成本主要来自上下文长度、模型推理、工具或 API 调用、运行频率和多角色并行。控制成本先从任务设计入手：只提供当前需要的资料，固定输出结构和长度，简单分类或提取使用响应更快的方案，复杂推理再用更强模型；稳定子流程用 Workflow 或脚本，不让 Agent 每次重新思考。自动化设置批次 ID 防重复，记录单次 Token、API 次数和耗时，持续观察趋势。若成本超预算，应暂停下一轮并由 owner 确认，而不是悄悄降低质量或无限重试。

89. 为什么要保留问题复现材料？

答：问题报告应当让收到的人可以直接行动。至少记录发生时间、系统与应用版本、任务原文、工作模式、目标目录或连接器、输入文件名、预期结果、实际现象、完整报错和复现步骤；再附关键截图、日志、批次 ID 和已生成文件路径。涉及远程任务时补充本机或云端模式、目标

电脑与账号状态；涉及外部平台时说明授权和错误码。提交前先脱敏，不发送密码、密钥和客户原文。这样的材料能迅速区分安装、网络、权限、数据源、Skill 或输出问题。

90. 正式交付前要做哪些检查？

答：正式交付可以分四层验收。内容层核对事实、数字、引用和待确认项；结构层检查标题、字段、页数、责任人和决策项是否齐全；文件层确认名称、版本、路径、可编辑性、公式和链接；视觉与安全层逐页查看表格、图片、分页、投屏效果及敏感信息。对外发送前再确认共享范围、版权和批准人。若由多个 Agent 生成，还要检查术语与主版本一致。最后保存源文件、阅读版和验收记录，不能因为对话预览正常就省略对实际交付物的打开检查。验收人和验收时间也应留痕。

A.10 复用进阶与团队应用（91—100）

91. 怎样把一次成功任务变成模板？

答：从 Prompt 到任务卡、SOP 和 Skill，模板化的第一步是从成功任务中提取稳定结构，而不是复制整段聊天。保留目标、输入、动作、约束、输出和验收六项，把人名、路径、日期、指标和受众改成清晰占位符；同时写明适用与不适用场景、主版本规则和资料缺失时的处理。再用与原案例不同的一组真实或脱敏材料测试，观察哪些步骤仍需临时解释。若模板只能复现原案例，就继续修改；通过多次测试后，才适合进入团队模板库，并附版本、样例和变更说明。

92. 什么时候把模板升级成 Skill？

答：升级条件不是“Prompt 很长”，而是同类任务已经重复成功，步骤、工具、业务规则、异常分支和输出格式相对稳定，并且复用价值高。升级时将触发条件和主流程写入 SKILL.md，业务口径放 references，可重复操作放 scripts，固定版式放 templates，正常、缺字段、冲突和工具失败场景放 tests。先在隔离目录压力测试，确认需要权限和网络时会明确提示，动作不可逆时会停下。发布后记录版本与维护人，流程改变就回归测试，不把 Skill 当作一次编写、永久正确的黑箱。

93. 如何建立自己的高频任务库？

答：先连续两到四周记录真实工作，不要凭印象列“想自动化的事”。每项任务登记发生频率、人工耗时、输入来源、是否有固定步骤、结果能否验收、错误影响和所需权限。优先选择高频、低风险、资料稳定的事项，如固定格式整理和周期汇总；高风险但低频的工作先保留人工。每个候选任务保存原始 Prompt、输入样例、合格产物、验收规则、常见失败和人工修复方法。跑通后先形成任务卡，再按复用程度升级为 Skill 或自动化。任务库的价值是减少重复试错，不是展示功能数量。

94. 为什么要给模板和 Skill 做版本管理？

答：Skill 文件化后可以用 Git 管理、版本回滚、团队共享、A/B 测试和持续更新。业务口径、模型、外部接口、模板和权限都可能变化，若没有版本记录，同一任务结果变了就无法判断原因。每次更新至少写日期、作者、改动内容、适用范围、依赖变化和测试结果；重大改动提升主版本，小修订也保留变更说明。旧版移入归档并标明失效，不要与当前版同时默认触发。上线前用固定样本做回归，失败时能够安全回退，而不是在生产任务中临时修补。

95. 怎样让 WorkBuddy 更懂自己的偏好？

答：记忆可以跨任务保存偏好、长期规则、项目决策或历史状态，但错误记忆也可能被反复使用。适合保存的是稳定、低敏感、可明确表达的偏好，如称呼、语言、标题层级、常用文件格

式和审稿规则；重要项目事实则应放有来源和版本的文件，不只依赖记忆。不要保存密码、密钥、客户隐私和短期口径。每条重要记忆最好有日期、来源和更新机制，发现错误立即纠正或删除。自动重组和更新不代表每条记忆都始终有效，因此还要定期审查，而不是越多越好。长期偏好也要设置复核周期。

96. 新人如何在七天内入门？

答：七天入门可以安排成一条闭环路线：第一天从官方入口安装、登录并建立演练目录；第二天认识侧边栏、对话区、结果区，以及默认模式、Ask 和 Plan；第三天练文件读取并用清单校验；第四天练六要素任务说明；第五天完成会议纪要或 Excel 小任务；第六天尝试一个来源可信的 Skill 或只读连接器；第七天复盘输入、产物、错误和验收。每次只扩大一个变量，处理公开或脱敏材料。七天目标不是掌握所有入口，而是能安全完成一个可追溯的真实任务。

97. 从熟练使用到进阶的关键是什么？

答：真正的进阶标志，是能正确组合模型、Agent、Skill、工具、知识库和 Workflow，而不是记住更多按钮。先把事实资料放进有版本的知识库，把稳定做法封装为 Skill，把长期目标和状态放进项目；路径固定的子流程用 Workflow，路径不确定的任务再交给 Agent，重复触发且可靠性达标后才做自动化。每一层都设置最小权限、检查点、日志、停止条件和人工验收。这样发生错误时能知道是资料、方法、连接还是控制流出了问题，系统才可维护、可复用、可扩展。

98. 团队推广 WorkBuddy 应该从哪里开始？

答：团队推广应从一个高频、低风险、结果可量化的场景试点，例如已脱敏资料问答、会议纪要草稿或固定报表。先明确 owner、主版本、输入与验收标准，只给小范围成员和最小权限；记录人工耗时、错误、返工、采用率和成本。试点稳定后，补充运行手册、成员权限、备份处理人和变更流程，再逐步扩大。不要把一个人的成功 Prompt 直接推给全公司，也不要一开始连接所有生产系统。团队落地的核心是统一口径与责任，而不是安装数量。试点通过后再扩大应用范围。

99. 如何判断自动化真的提升了效率？

答：自动化上线后要长期观察按时触发率、一次运行成功率、数据源可用率、有效结果数量、推送成功率和单次运行成本。业务层还要比较上线前后的人工耗时、返工次数、错误率、采用率和决策时效，并把维护 Prompt、处理告警和人工审核的时间算进去。若内容生成很快，却需要大量纠错或经常重复推送，不能算真正提效。指标持续下降时，先定位是数据源、权限、模型、流程还是验收门禁变化；每次修改后先手动验证三次，再更新定时配置，避免直接在生产自动化上实验。

100. 学会 WorkBuddy 的最终标准是什么？

答：真正会用 WorkBuddy 的标准可以看成五件事：能把工作写成包含目标、输入、动作、约束、输出和验收的任务；能建立清楚的工作目录和主版本；能判断何时用 Skill、连接器、知识库、Agent 或固定 Workflow；能通过来源、日志和实际文件验收结果；能在缺资料、越权限、超预算或不可逆动作前让系统停下。进一步还能把重复成功的方法沉淀为模板和 Skill，把可靠流程升级为自动化，并持续复盘。功能会变，负责地交付结果的能力才是稳定标准。